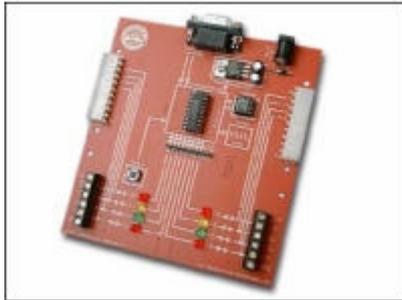


## **PICAXE-18M2-ST Microcontroller chip (Product Code: STP-018)**

The PICAXE-18M-ST is a special version of the PICAXE-18M2 firmware that is primarily designed to convert the 'Stamp Controller' project board used in Technological Studies courses in Scotland to PICAXE use.



To upgrade the system simply remove the old 18 pin chip in the centre of the Stamp Controller and replace with the new PICAXE-18M2-ST chip, ensuring pin1 is at the top of the board. That's it – easy to do in under 30 seconds!

Also make sure you have downloaded the latest version of the PICAXE Programming Editor software (5.5.0 or later) from [www.picaxe.com/PE](http://www.picaxe.com/PE)

By simply replacing the existing chip on the Stamp Controller with the new PICAXE-18M2-ST chip the entire system is then converted to a 'PICAXE-18M2' system, so can be used with any software that supports the PICAXE system. The free BASIC language software and Stamp Controller hardware can then easily be used with all the modern Windows versions such as Vista and 7, as well as Mac and Linux computers.

The 18M2-ST upgrade also provides a large range of improvements such as parallel task processing, additional variables and analogue inputs.

### **Quick Comparison**

Almost every feature on the PICAXE-18M2-ST is enhanced, for instance programs can be up to 8x longer than before.

<b>Feature</b>	<b>18M2ST</b>	<b>PBASIC1.4</b>
Memory Capacity	<b>2048 bytes Internal</b>	256 bytes External
EEPROM Capacity	<b>256 bytes Internal</b>	Shared with program
RAM (b variables)	<b>28 (b0-b27)</b>	14 (b0-b13)
Total RAM	<b>512 bytes</b>	14 bytes
Parallel Tasks	<b>Yes, up to 8 tasks</b>	No
Resonator	<b>Up to 32MHz, internal</b>	4MHz, external
General I/O pins	<b>13</b>	8
10-bit ADC/Touch pins	<b>10</b>	0
Ring tone support	<b>Yes</b>	No
I2C / Infrared	<b>Yes</b>	No
Serial over download cable	<b>Yes</b>	No
Programming (BASIC)	<b>Yes</b>	Yes
Programming (Flowcharts)	<b>Yes</b>	No
Windows, Mac and Linux	<b>Yes</b>	No
Operating Voltage	<b>1.9 to 5.5V</b>	4.5 to 5.5V

For PICAXE manuals, software downloads and further information see the documentation for the PICAXE-18M2 at [www.picaxe.com](http://www.picaxe.com)

## **Frequently Asked Questions**

### **What are the physical differences between the 18M2 and the 18M2-ST chips?**

- The 18M2-ST has the download pins 'polarity inverted' for compatibility with the Stamp Controller project board. This change is 'invisible' to the end users program, so in effect can be ignored whilst programming.
- The 18M2-ST has a physical active-low 'reset' pin on leg 4.
- The 18M2-ST is supplied on a chip labelled 'PIC16F1847'
- The free Programming Editor software does not see any difference at all between the two parts. Therefore simply select the 'PICAXE-18M2' mode within the PICAXE software.

### **How are the i/o pins allocated on the Stamp Controller?**

- The 8 i/o pins are PICAXE portB, so are pins B.0 to B.7
- All portB pins can be inputs or outputs (use 'dirsB=' to configure)
- All portB pins except B.0 can also be used as analogue inputs
- To switch an output on use the terminology 'high B.7'
- To use as an input use the terminology 'if pinB.0 = 1 then'
- To use as an analogue input use the terminology 'readadc B.2, b0'
- The resonator/EEPROM pins are on portA, and may now also be used as extra general purpose i/o pins if required.

### **How do I convert the 9 pin serial connector on the Stamp Controller to USB?**

- Simply use the low cost USB<>serial adapter, part USB010
- If your computer still has traditional 9 way serial ports you can still use the older style serial cable with the PICAXE-18M2-ST

### **Do I need the external EEPROM and resonator for the 18M2-ST?**

- No, these features are now 'internal' on the 18M2-ST and so these external devices can be removed (or simply just left in position and not used).
- If desired the pins previously used for the features may be used as general purpose i/o pins (portA)

### **Can I use a PICAXE-18M2-ST on a normal PICAXE project?**

- No, the 18M2-ST is only for boards originally designed for use with the PBASIC1.4 chip, such as the Stamp Controller. The 18M2-ST will not function on a PICAXE board with an AXE027 PICAXE USB cable. For normal PICAXE boards simply use the normal PICAXE-18M2 chip instead!

### **Why does the reset pin not work?**

- The 18M2-ST does have a reset pin, which will reset the chip when brought to 0V. However some Stamp boards have an external 'brownout' circuit built with transistors that does not allow the voltage to drop quite low enough for the 18M2-ST reset pin to activate. In this case replace the whole external brownout circuit with a simple 4k7 pull up resistor and switch to 0V (18M2-ST chips have an internal brownout circuit) .

### **How do I use the Higher Still Stamp Controller Input Module?**

- The Input Module has 4 digital inputs, now connected to the STP018 pins B.0 to B.3. Due to the pull down resistor on the Stamp Controller switches can be connected directly between the two contacts, and used with syntax such as  
if pinB.0 = 1 then...
- Digital inputs B.1 to B.3 (not B.0) now also have direct internal analogue capabilities due to the improved functionality of the STP018 chip. So you can also use an LDR directly between the two contacts on the digital input and use code such as this (with slide switch in top position):  
readadc B.1, b1 ; read analogue value on pin B.1 into variable b1  
if b1 > 100 then...
- The old PBASIC1.4 chip had no internal analogue capabilities at all, so the purpose of the ADC PIC chip on the Inputs Module was to convert analogue values into different length pulses (which could then be read by extended BASIC 'sensorA' type commands, which actually internally 'faked' a pulsins command to read the pulses).

So to use the old wired LDR/thermistors via the 3 way connectors use the following equivalent code (with slide switch in bottom position):

```
symbol sensorA = b11  
symbol sensorB = b12
```

```
pulsins B.2, 1, sensorA ; read pulse on B.2 into variable  
if sensorA > 100 then...
```

```
pulsins B.3, 1, sensorB ; read pulse on B.3 into variable  
if sensorB > 100 then...
```

### **How do I use the Higher Still Stamp Controller Output Module?**

- No change, use as before, but it is recommended to change the keyword 'dirs' to 'dirsB' and 'pins' to 'pinsB' within your programs.  
let pinsB = %11110000

### **How do I use the Higher Still Stamp Controller LCD/Printer Modules?**

- No change, use as before.